Artistic Style transfer

Sameer
Indian Institute of Technology Ropar
Ropar, Punjab
2016csb1058@iitrpr.ac.in

Vineet
Indian Institute of Technology Ropar
Ropar, Punjab
2016csb1063@iitrpr.ac.in

Ram Krishna
Indian Institute of Technology Ropar
Ropar, Punjab
2016csb1053@iitrpr.ac.in

ABSTRACT

Rendering the semantic content of an image in different styles is a difficult image processing task. Arguably, a major limiting factor for previous approaches has been the lack of image representations that explicitly represent semantic information and, thus, allow to separate image content from style. Here we use image representations derived from Convolutional Neural Networks optimised for object recognition, which make high level image information explicit.We implemented Neural Algorithm of Artistic Style and its two variations - Combining Markov Random Fields and Semantic Style Transfer.

ACM Reference Format:
Sameer, Vineet, and Ram Krishna. 2018. Artistic Style transfer . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Image processing algorithms have improved dramatically thanks to CNNs trained on image classification problems to extract underlying patterns from large datasets [8]. As a result, deep convolution layers in these networks provide a more expressive feature space compared to raw pixel layers, which proves useful not only for classification but also generation [7]. For transferring style between two images in particular, results are astonishing especially with painterly, sketch or abstract styles [3].

However, to achieve good results using neural style transfer in practice today, users must pay particular attention to the composition and/or style image selection, or risk seeing unpredictable and incorrect patterns. For portraits, facial features can be ruined by incursions of background colors or clothing texture, and for landscapes pieces of vegetation may be found in the sky or other incoherent places. There is certainly a place for this kind of glitch art, but many users become discouraged not being able to get results they want. Through our social media bot that first provided these algorithms as a service [1], we observe that users have clear expectations how style transfer should occur: most often this matches semantic labels, e.g. hair style and skin tones should transfer respectively regardless of color. Unfortunately, while CNNs routinely extract semantic information during classification, such information is poorly exploited by generative algorithms as evidenced by frequent glitches. We attribute these problems to two underlying causes:

(1) While CNNs used for classification can be re-purposed to extract style features (e.g. textures, grain, strokes), they were not architected or trained for correct synthesis.
(2) Higher-level layers contain the most meaningful information, but this is not exploited by the lower-level layers used in generative architectures: only error backpropagation indirectly connects layers from top to bottom.

To remedy this, Semantic Style transfer [2] algorithm introduce an architecture that bridges the gap between generative algorithms and pixel labeling neural networks. The architecture commonly used for image synthesis [8] is augmented with semantic information that can be used during generation. Then we explain how existing algorithms can be adapted to include such annotations, and finally we show-case some applications in style transfer as well as image synthesis by analogy.

2 RELATED WORKS

The image analogy algorithm [4] is able to transfer artistic style using pixel features and their local neighborhoods. While more recent algorithms using deep neural networks generate better quality results from a stylistic perspective, this technique allows users to synthesize new images based on simple annotations. As for recent work on style transfer, it can be split into two categories: specialized algorithms or more general neural approaches. The first neural network approach to style transfer is gram-based [3], using so called Gram Matrices to represent global statistics about the image based on output from convolution layers. These statistics are computed by taking the inner product of intermediate activations,a tensor operation that results in a N into N matrix for each layer of N channels. During this operation, all local information about pixels is lost, and only correlations between the different channel activations remain. When glitches occur, it is most often due to these global statistics being imposed onto the target image regardless of its own statistical distribution, and without any understanding of local pixel context. A more recent alternative involves a patch-based approach [6], which also operates on the output of convolution layers. For layers of N channels, neural patches of 3 into 3 are matched between the style and content image using a nearest neighbor calculation. Operating on patches in such a way gives the algorithm local understanding of the patterns in the image, which overall improves the precision of the style transfer since fewer errors are introduced by globally enforcing statistical distributions. Both gram- and patch-based approaches struggle to provide reliable user controls to help address glitches. The primary parameter exposed is a weighting factor between style and content; adjusting this results in either an abstract-styled mashup that mostly ignores the input content image,

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

or the content appears clearly but its texture looks washed out (see Figure 2, second column). Finding a compromise where content is replicated precisely and the style is faithful remains a challenge in particular because the algorithm lacks semantic understanding of the input. Thankfully, recent CNN architectures are capable of providing such semantic context, typically by performing pixel labeling and segmentation [9]. These models rely primarily on convolutional layers to extract high-level patterns, then use deconvolution to label the individual pixels. However, such insights are not yet used for synthesisâĂŤ despite benefits shown by non-neural approaches. The state-of-the-artspecialized approaches to style transfer exploit semantic information to great effect, performing color transfer on photo portraits using specifically crafted image segmentation [10]. In particular, facial features are extracted to create masks for the image, then masked segments are processed independently and colors can be transferred between each corresponding part (e.g background, clothes, mouth, eyes, etc.) Thanks to the additional semantic information, even simpler histogram matching algorithms may be used to transfer colors successfully

3 METHODOLOGY

The *neural style transfer* is the art of modern deep learning.The Major terms used in this literature are the content image(refer to image to which style is applied),style image(the image whose style is applied) and regularization term is added in context of smoothening the images.The three implementations are presented in the report all are implemented in tensorflow and keras.The basic network used for the models is the pre-trained VGG19 network(the Imagenet-2014 winner) [8].In each implementation random noise is taken as the initial image and fed into network.In each iteration, input image change by back propagation on the defined loss in the respective implementation.

3.1 Neural Algorithm of Artistic Style

In this algorithm[3], content loss and style are defined as described in the below subsections.The total loss is weighted sum of the content loss and style loss.Gradient Descent is used for back propagation along with adam-optimizer on the total loss.In each iteration , back propagation is done w.r.t the input image(generated image(x)) and the weights of the VGG19 network does not change.

3.1.1 Content Loss. The generated image(x) and content image(c) are feed into the VGG19 network and then their output at selected content layers are considered for the content loss. Content loss is basically the squared loss of the layer outputs of $\mathbf{x}$ and $\mathbf{c}$.

3.1.2 Style loss. The output for the generated image(x) and style image(s) is taken at selected layers called as style layers. Instead of squared loss of the outputs , the squared loss of the their Gram-Matrix is taken as the Style Loss.

3.2 Combining Markov Random Fields

In this implementation, the dCNN network, optimizer and the content loss is taken same as in Neural algorithm but the style loss is computed by using Markov Random Fields on the style layers of the network and a new regularizer term is added to restrict the back propogation. These changes are described in the below subsections-

Figure 1: Style Image and Content Image

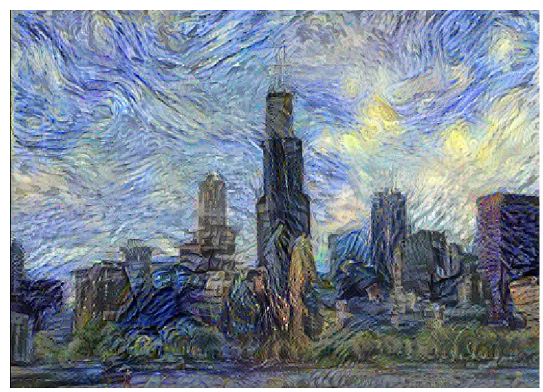

Figure 2: Generated Image by neural style transfer (Gateyes) 1000 iterations

3.2.1 Style Loss. The output for the generated image(x) and style image(s) is computed at style layers. Each layer output is broken into patches of size (3,3).It is done for both generated image(x) and style image(s) outputs. At each layer , the square difference of the patch of input image and the corresponding nearest-neighbor patch of the style is image is taken as Style loss[6].Nearest neighbor patch is found by using normalized cross-correlation over all patches of the style image.

3.2.2 Regularizer. There is significant amount of low-level image information discarded during the discriminative training of the network. In consequence, reconstructing an image from its neural encoding can be noisy and unnatural. For this reason, we penalize the squared gradient norm[6] to encourage smoothness in the synthesized image.
The synthesized image is shifted by 1 pixel in the right and down

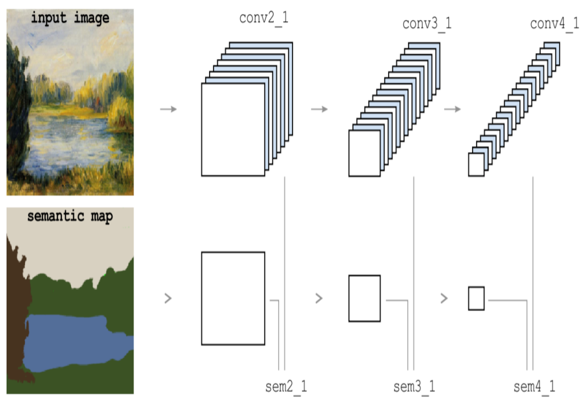

Figure 3: Content image and its corresponding semantic image output at different layers.

direction. Regularized term is calculated by taking the difference of the synthesized and these two shifted synthesized image respectively.

3.3 Semantic Style Transfer

In neural style transfer , facial features can be ruined by incursions of background colors or clothing texture, and for landscapes pieces of vegetation may be found in the sky or other incoherent places. To tackle these bad results, we take semantic image of the style and the content image along with them as shown in the figure below.The algorithm is almost same as used in Combining Markov Random Fields implementation but while extracting patches at each layer, the output of the semantic image of the content image is concatenated at the end of the output of the input image and the output of the semantic image of the style image is concatenated at the end of the output of the style image is described in the Semantic Style transfer[2]

4 EXPERIMENTATION

The model data is taken from the repository of kaggle (painter-by-numbers), wikipedia images and images provided in our reference model repository.The model can be evaluated using some general guidelines like comparing the images obtained with different models available online. There is no deterministic way to evaluate the model but the factors like quality of image,artifact features and the style implementation are used to evaluate the models. The models perform variably upon different images and there is no certain winner in this case.

However increasing the number of iterations generally gives better output. Also when we change the content weight to too high then the output image contains major part of the content image than the style image.

5 RESULTS AND DISCUSSION

The same content and style image is given as input in the three implementations as shown in the figures below. The final image by MRF and semantic style implementation creates better image than neural style transfer implementation as in other two ,the loss is computed between the nearest matched patches due to which style features of whole image is not applied to a particular image segment and only the features of matched patches is applied to a particular image segment but it could be seen in the output of neural style the background color is affected by all the style features of input style image.

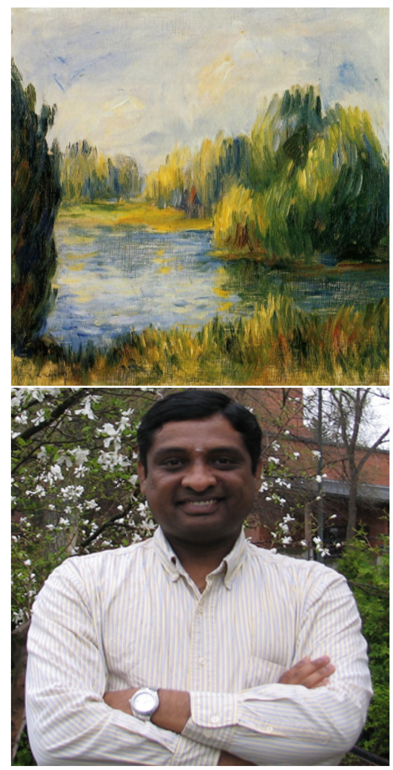

Figure 4: Style Image and Content Image

6 SUMMARY AND FUTURE WORK

The previous two style transfer techniques perform well when colors and/or accuracy do not matter too much for the output image (painterly, abstract or sketch styles, glitch art) or when both image patterns are already similar,which obviously reduces the appeal and applicability of such algorithms. In the third technique-semantic style transfer , these issues are resolved by annotating input images with a semantic map, either manually authored or from pixel labeling algorithms.

We can extend this project in future to apply neural style transfer in real time on mobile devices.In real time we need to make the computations faster. For faster computation[5], a network can be trained to predict the initial image instead of taking noise as initial image.

Figure 5: Generated Image by neural style transfer (Gateyes) 50 iterations

Figure 6: Generated Image by neural patches by combining MRF 5 iterations

REFERENCES

[1] A Champandard. 2015. Deep forger: Paint photos in the style of famous artists.

[2] Alex J. Champandard. 2016. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. *CoRR* abs/1603.01768 (2016).

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 327–340.

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. Springer, 694–711.

[6] Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2479–2486.

[7] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5188–5196.

[8] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[9] Martin Thoma. 2016. A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541* (2016).

Figure 7: Generated Image by Semantic style transfer 5 iterations

[10] Yue Yang, Hanli Zhao, Lihua You, Renlong Tu, Xueyi Wu, and Xiaogang Jin. 2017. Semantic portrait color transfer with internet images. *Multimedia Tools and Applications* 76, 1 (2017), 523–541.